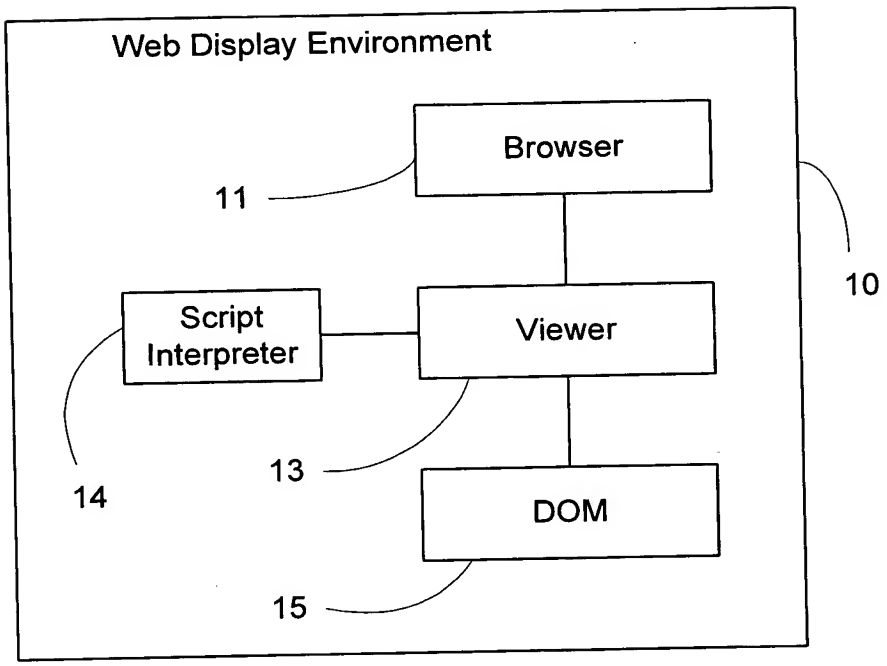



Figure 1
PRIOR ART

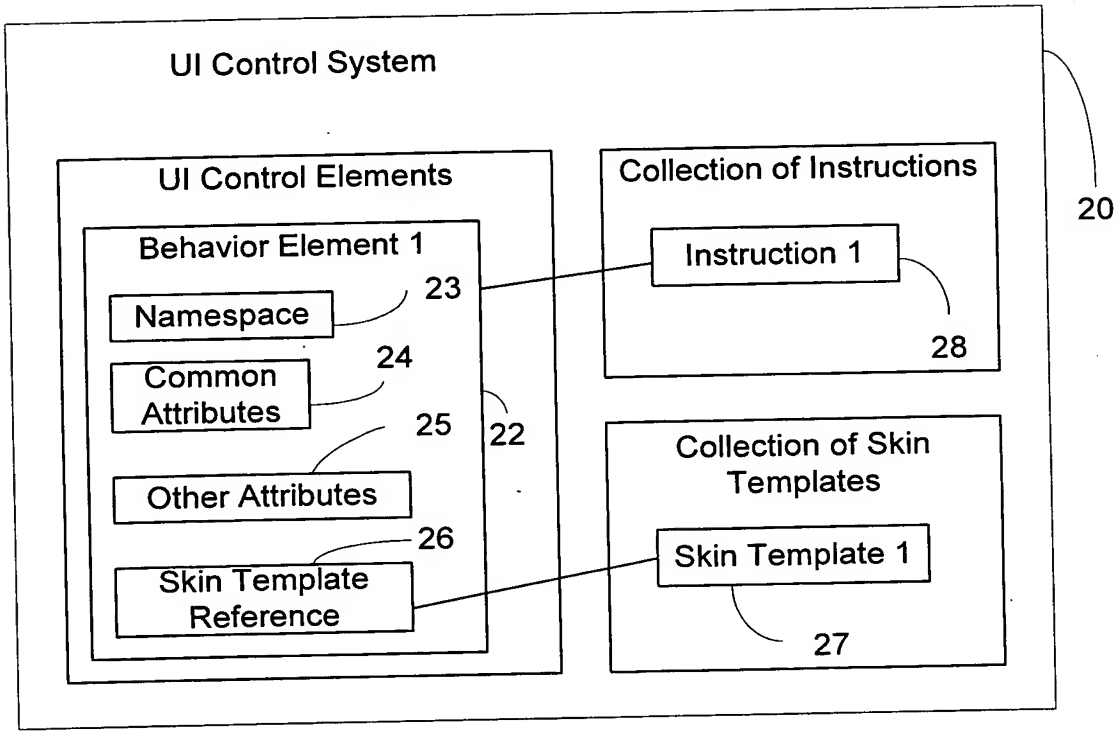



Figure 2

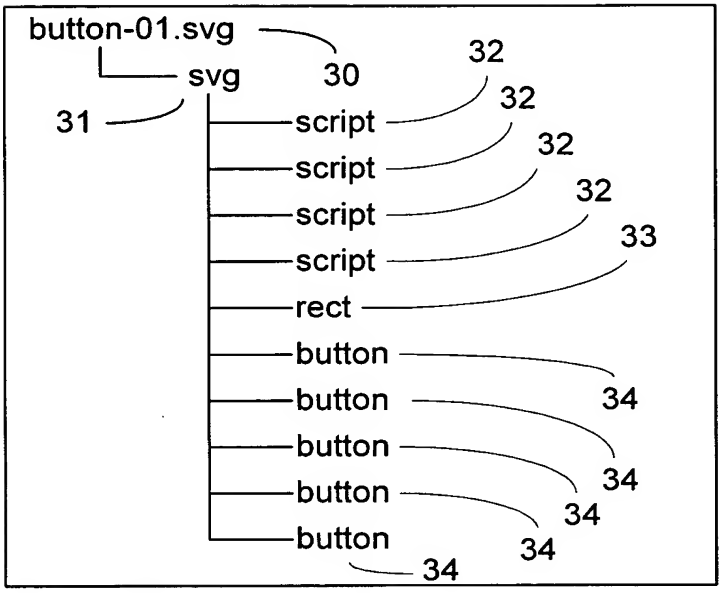

button-01.svg
30
svg
31
script
script
script
script
rect
button
button
button
button
32
32
32
32
33
34
34
34
34
34

Figure 3

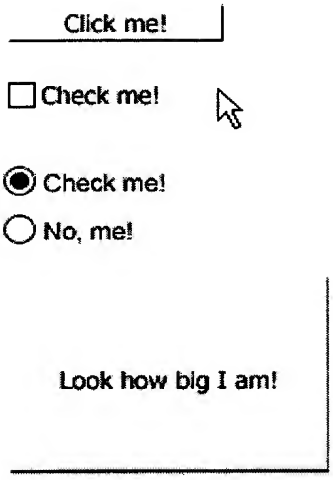



Figure 4A


Figure 4B

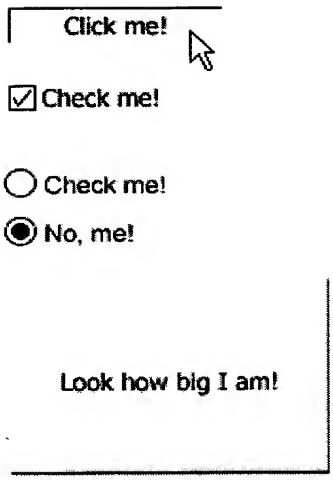



Figure 4C

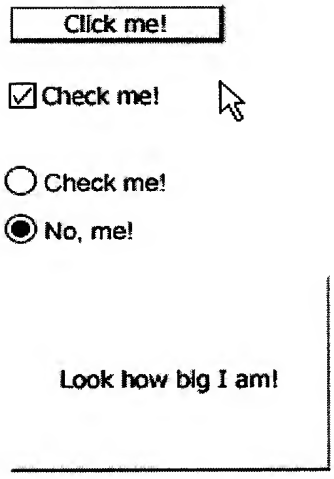



Figure 4D

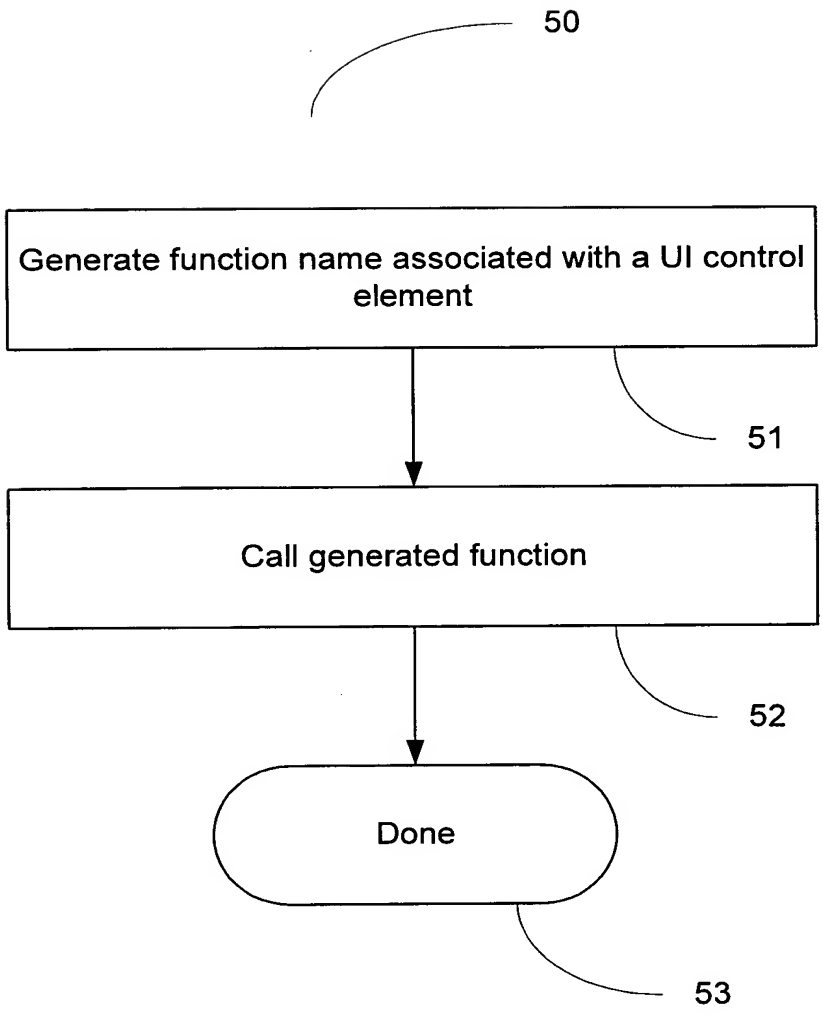

50
Generate function name associated with a UI control element
51
Call generated function
52
Done
53

Figure 5

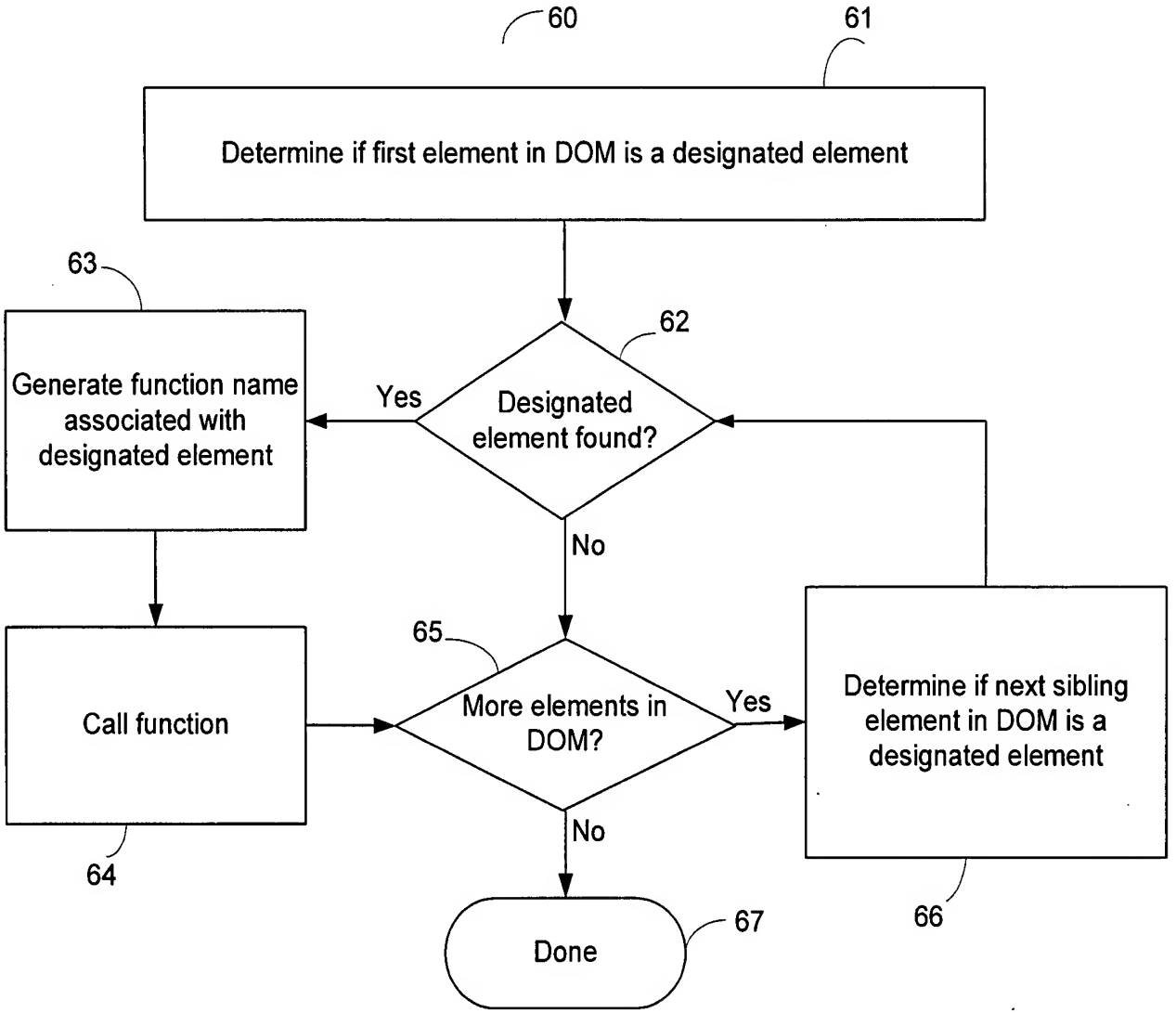

60
61
Determine if first element in DOM is a designated element
63
62
Generate function name associated with designated element
Yes
Designated element found?
No
65
Call function
More elements in DOM?
Yes
Determine if next sibling element in DOM is a designated element
No
64
Done
67
66

Figure 6

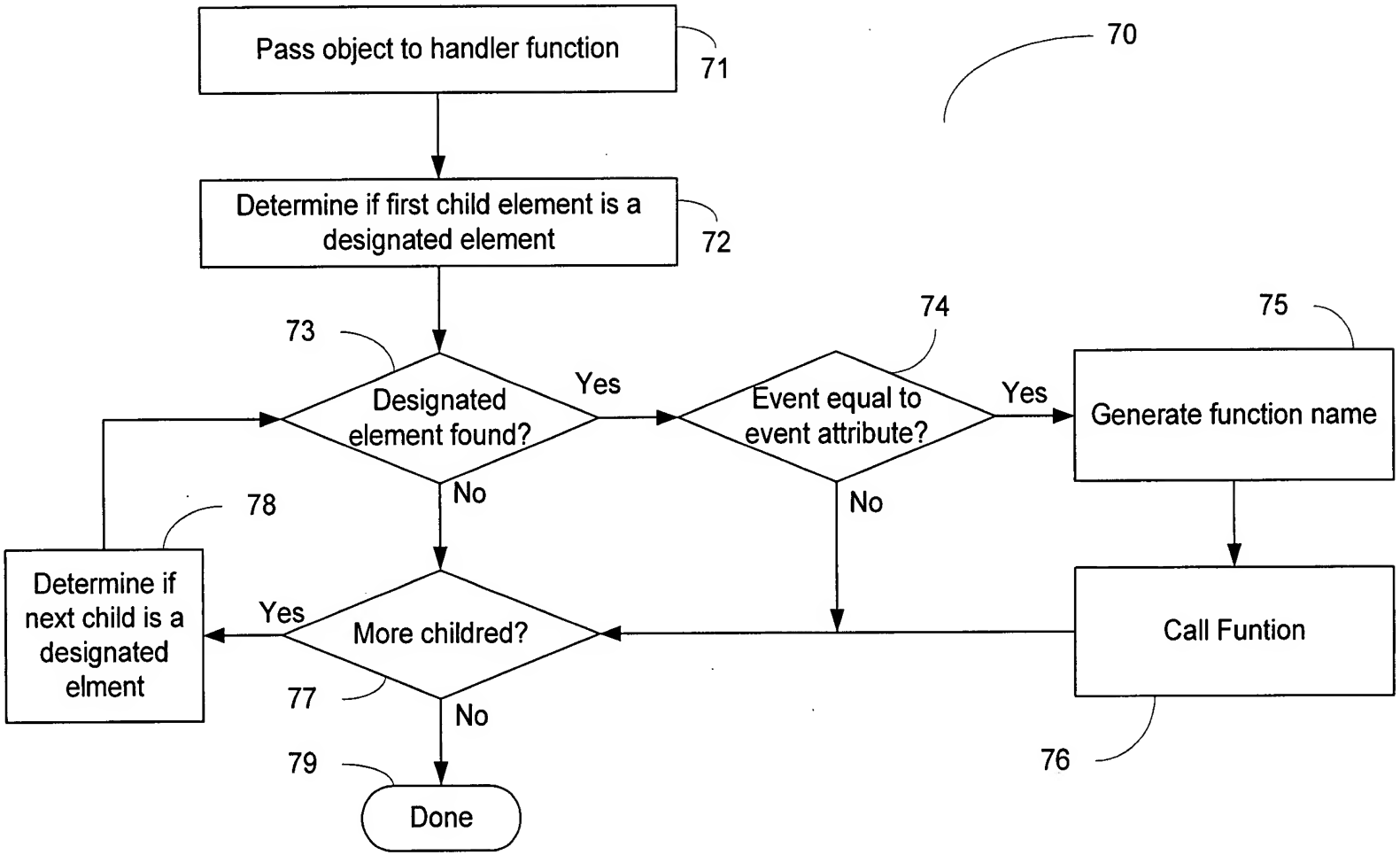

Pass object to handler function
71
70
Determine if first child element is a designated element
72
73
Designated element found?
Yes
74
Event equal to event attribute?
Yes
75
Generate function name
No
No
78
Determine if next child is a designated elment
Yes
More childred?
Call Funtion
77
No
76
79
Done

Figure 7

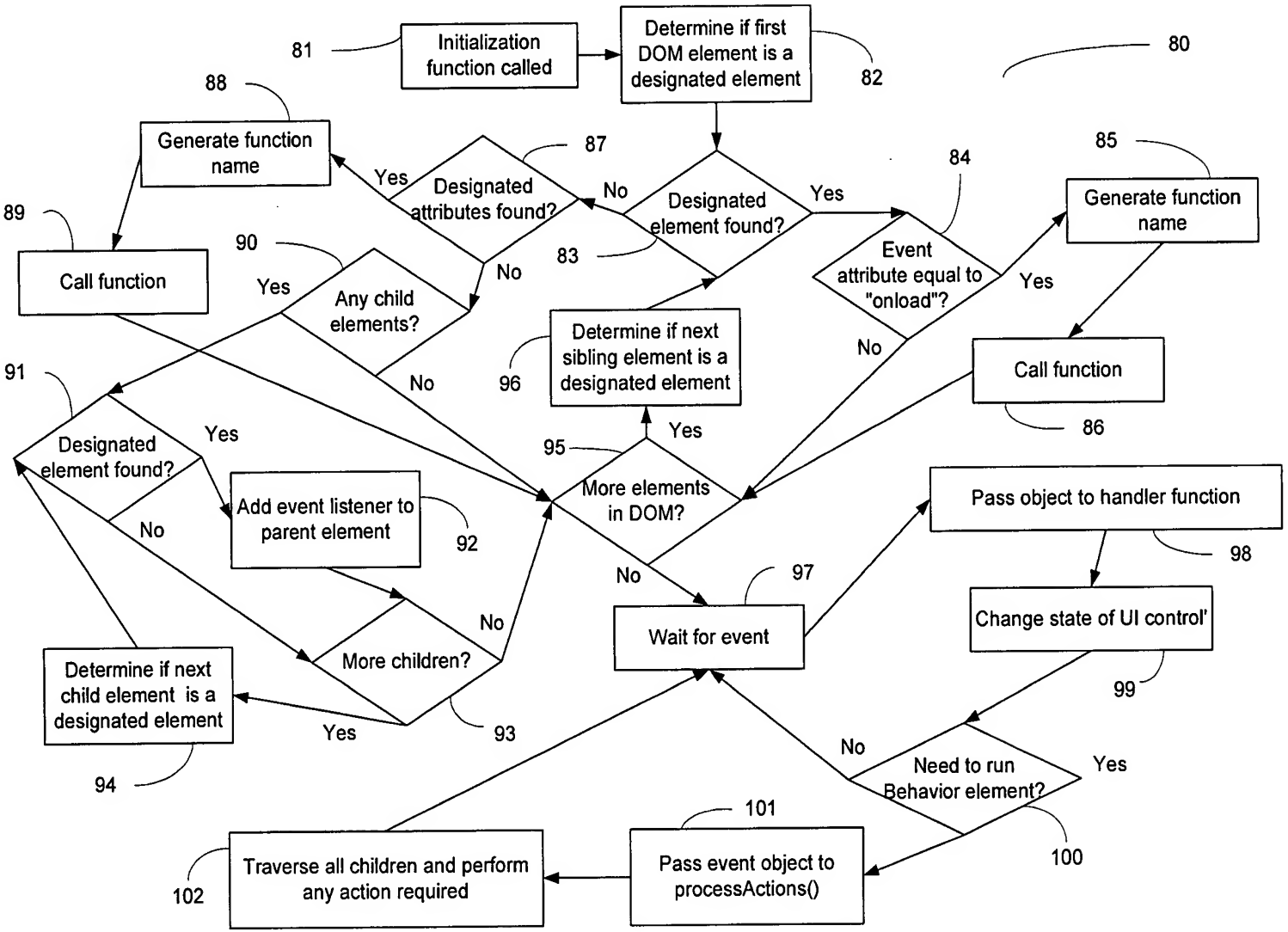

81
Initialization function called
Determine if first DOM element is a designated element
82
80
88
Generate function name
87
Yes
Designated attributes found?
No
Designated element found?
Yes
84
85
Generate function name
89
Call function
90
Yes
Any child elements?
No
83
No
Event attribute equal to "onload"?
Yes
91
Designated element found?
Yes
No
Add event listener to parent element
92
96
Determine if next sibling element is a designated element
No
Call function
86
95
Yes
More elements in DOM?
Pass object to handler function
98
No
97
Wait for event
Change state of UI control'
99
No
More children?
Yes
93
Determine if next child element is a designated element
94
Yes
No
Need to run Behavior element?
100
101
Pass event object to processActions()
102
Traverse all children and perform any action required

Figure 8

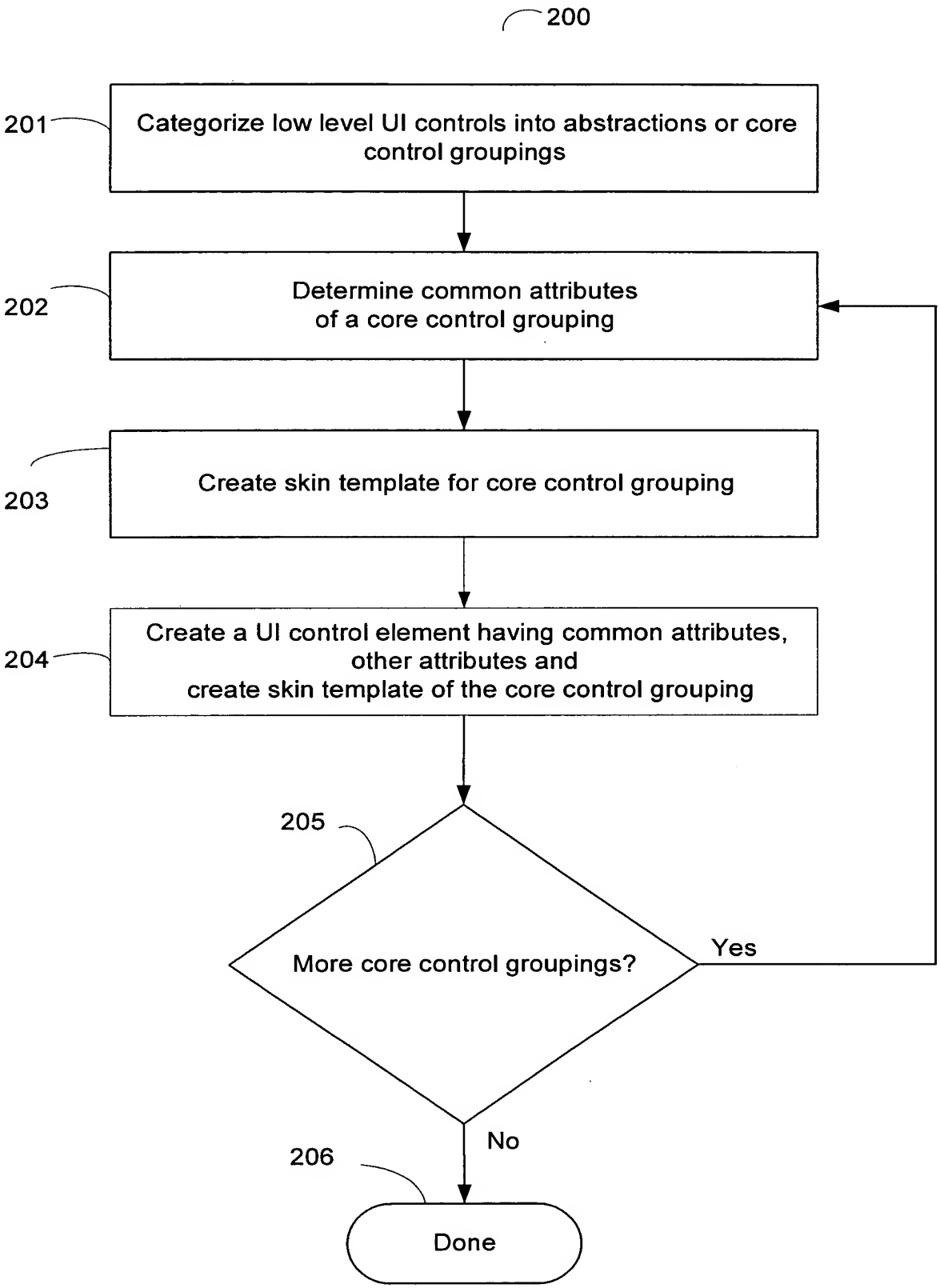

200
201
Categorize low level UI controls into abstractions or core control groupings
202
Determine common attributes of a core control grouping
203
Create skin template for core control grouping
204
Create a UI control element having common attributes, other attributes and create skin template of the core control grouping
205
More core control groupings?
Yes
No
206
Done

Figure 9

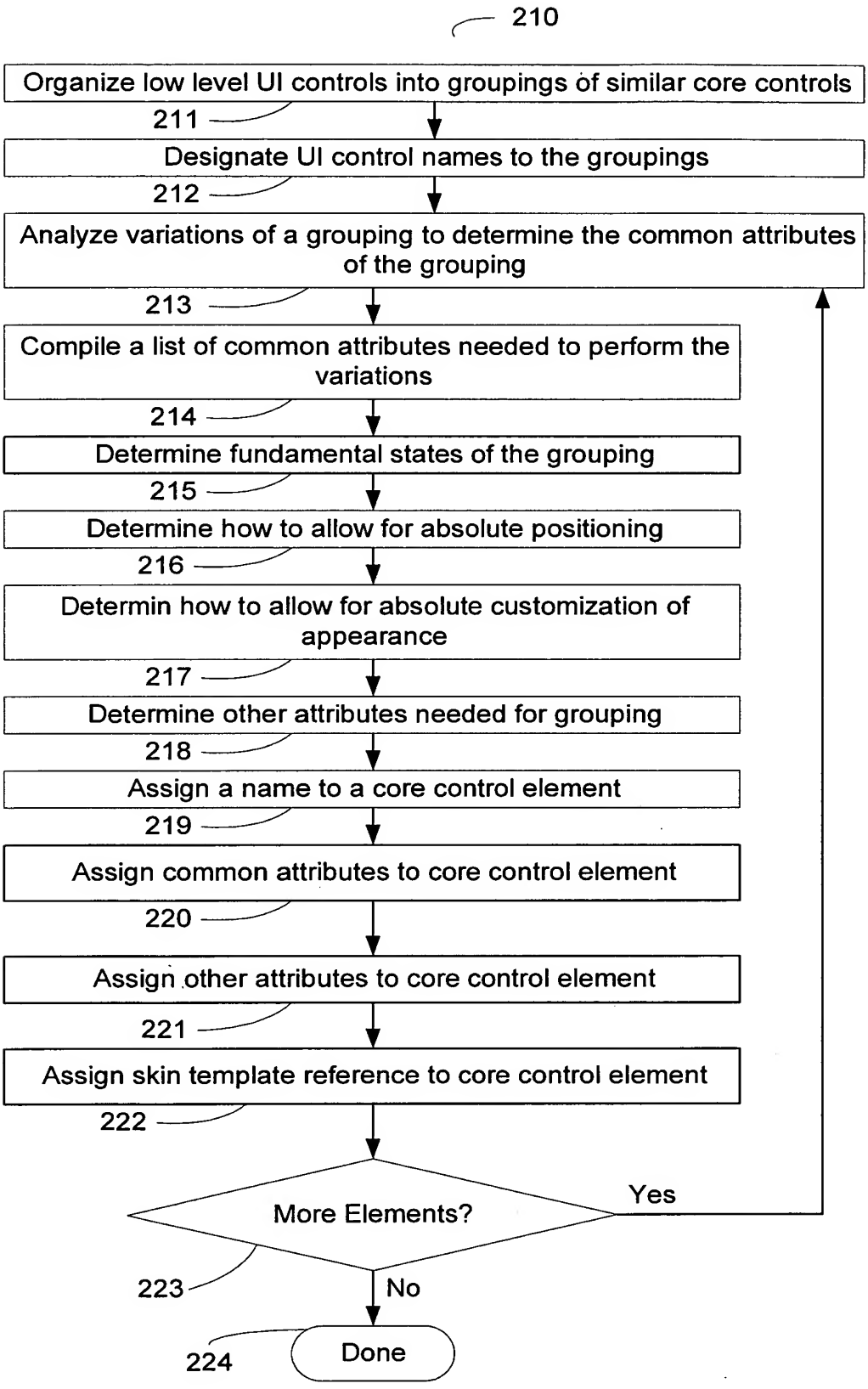
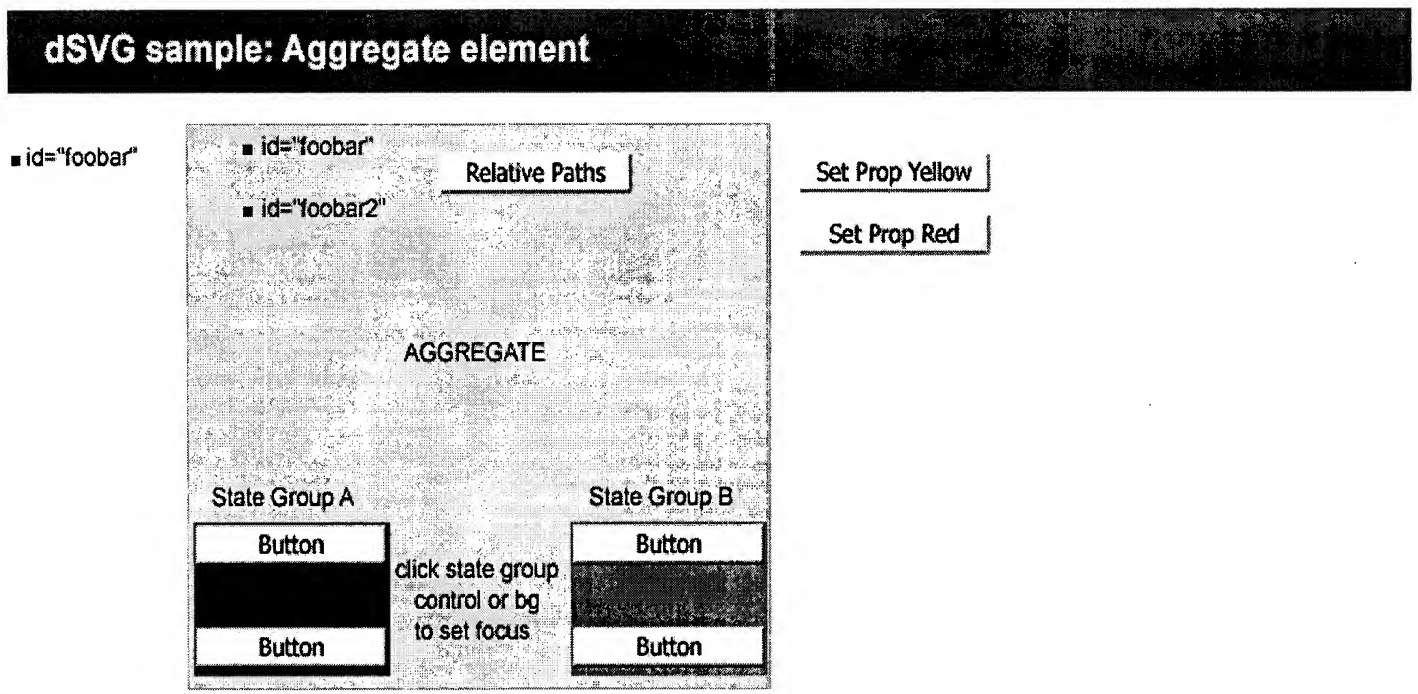

210
Organize low level UI controls into groupings of similar core controls
211
Designate UI control names to the groupings
212
Analyze variations of a grouping to determine the common attributes of the grouping
213
Compile a list of common attributes needed to perform the variations
214
Determine fundamental states of the grouping
215
Determine how to allow for absolute positioning
216
Determin how to allow for absolute customization of appearance
217
Determine other attributes needed for grouping
218
Assign a name to a core control element
219
Assign common attributes to core control element
220
Assign other attributes to core control element
221
Assign skin template reference to core control element
222
More Elements?
Yes
223
No
224
Done

Figure 10

## dSVG sample: Aggregate element




The aggregate element defines a top-level container element

In this sample, the buttons on the left and the blue rectangle are in stateGroup 'a' and the buttons on the right and the red rectangle are in stateGroup 'b'

The relative paths button drills into the alert statements inside itself to demonstrate use of 'this' keyword and into the fills of rectangles 'foobar' and 'foobar2' to demonstrate use of 'aggregate' keyword.

Figure 11

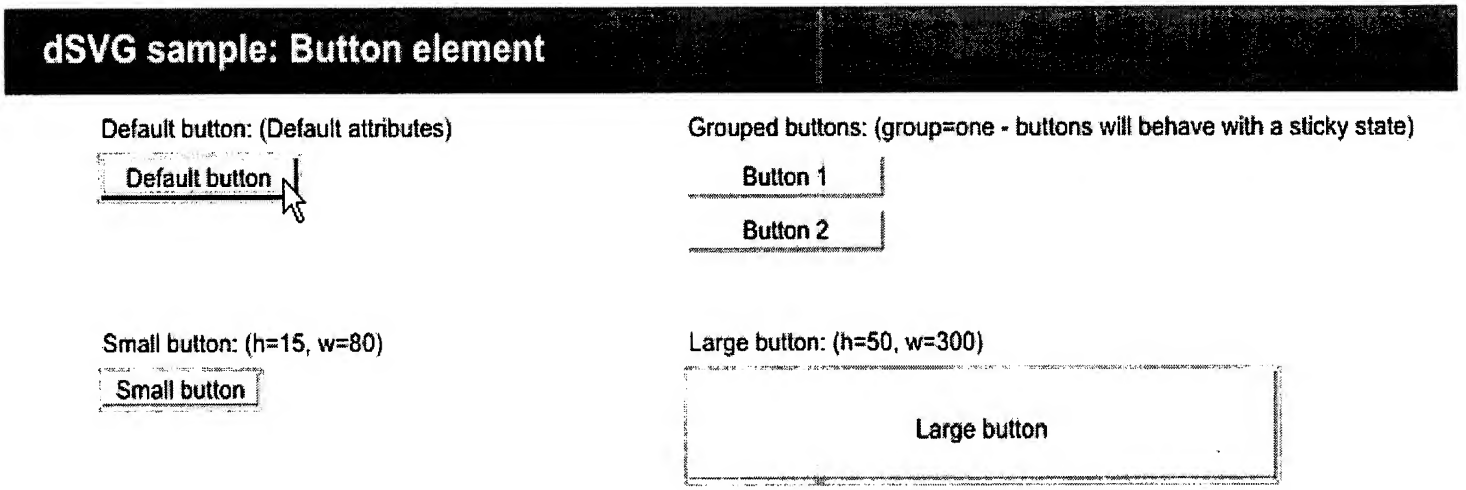

dSVG sample: Button element
Default button: (Default attributes)
Default button
Grouped buttons: (group=one - buttons will behave with a sticky state)
Button 1
Button 2
Small button: (h=15, w=80)
Small button
Large button: (h=50, w=300)
Large button
*Note: The red outlines are not part of the controls. They are used to identify the dimensions for each control.
Default button - This button has all of its attributes set to default values.
Grouped buttons - Buttons can belong to a group. If grouped, and toggle="true", this results in the functionality of a radio button.
Small and large buttons - Change the size of a control by specifying new values for the height and width attributes.

Figure 12

## dSVG sample: CheckBox element

Default check box: (h=12, w=12)

Small check box: (h=10, w=10)

Large check box: (h=40, w=40)

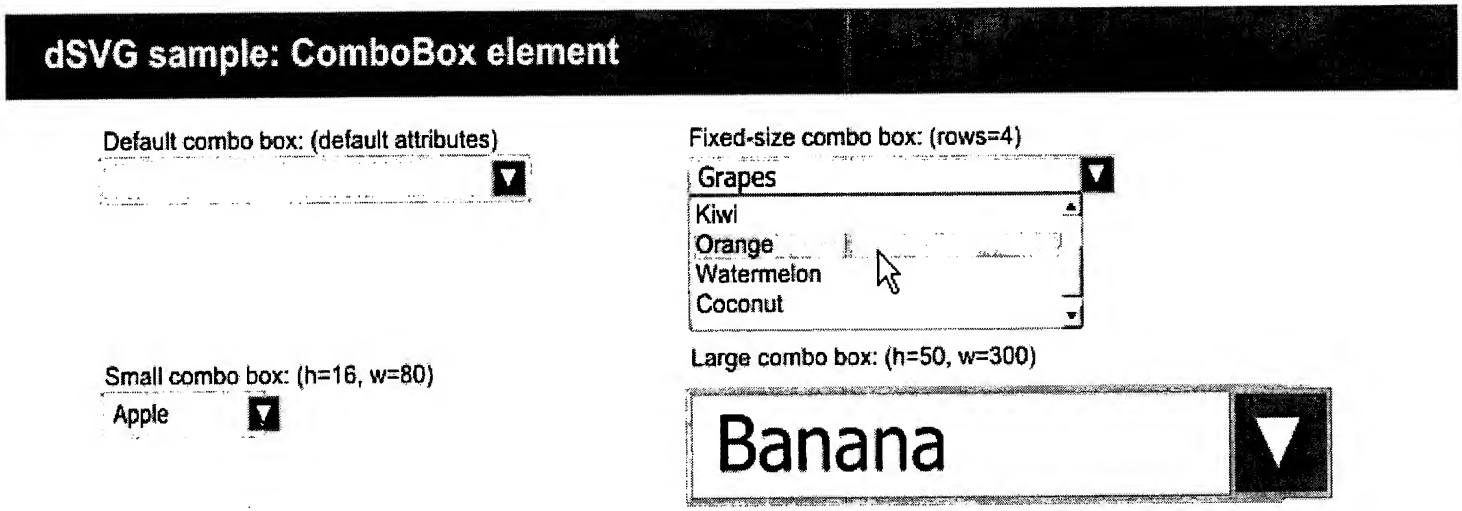
*Note: The red outlines are not part of the controls. They are used to identify the dimensions for each control.

---

Default check box - When you select the check box, it toggles between true (selected) and false (deselected).

Small and large check boxes - Change the size of a control by specifying new values for the height and width attributes.

Figure 13




*Note: The red outlines are not part of the controls. They are used to identify the dimensions for each control.

Default combo box - The default combo box has 3 child item elements: Apple, Banana, Grapes.

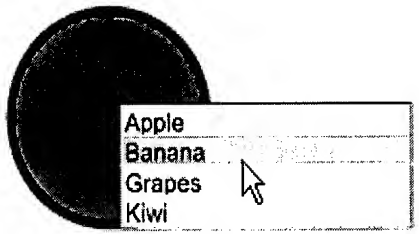
Fixed-size combo box - This combo box displays a maximum of 4 items. If the number of items exceeds 4, a scrollbar is enabled.

Small and large combo boxes - Change the size of a control by specifying new values for the height and width.

Figure 14

## dSVG sample: ContextMenu element

Default Context Menu: (default attributes) - right-click within the circle and the Context Menu should appear.



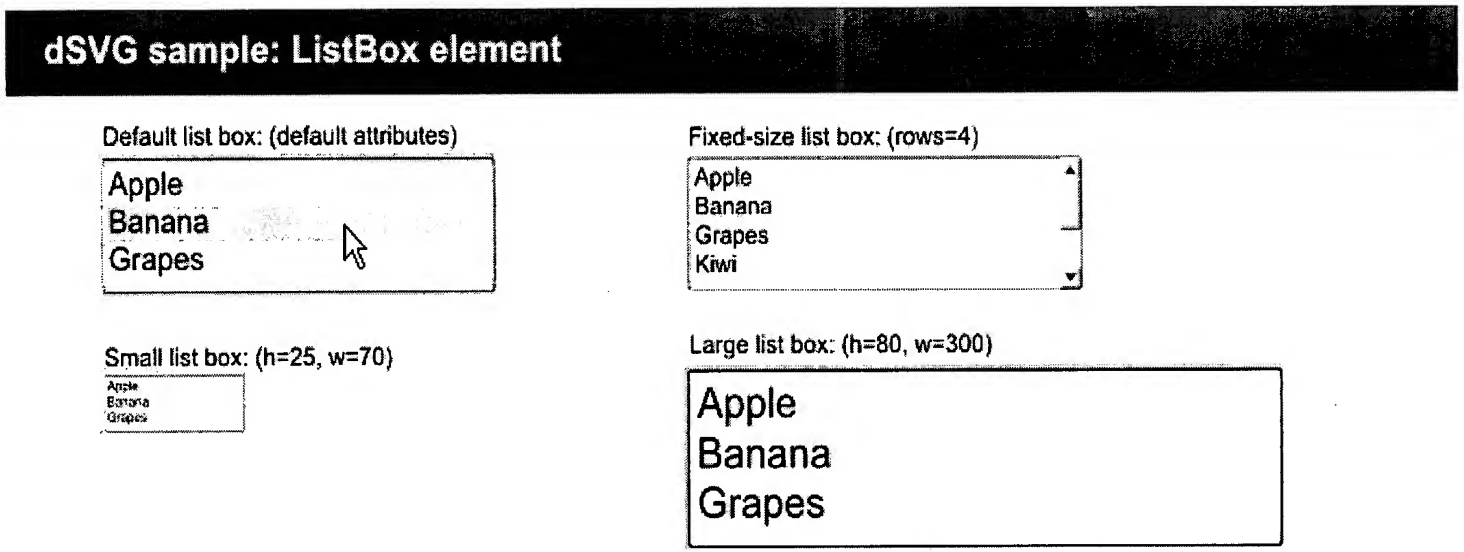

---

Default context menu - Right-click within the circle to display the context menu.

The list consists of items which can be added directly as child elements of the context menu.

The context menu is associated with the circle by adding a dsvg:contextMenu attribute to the circle which references the context menu.

Figure 15




Default list box - This list box has 3 child item elements: Apple, Banana, Grapes.

Fixed-size list box - This list box displays a maximum of 4 items. If the number of items exceeds 4, a scrollbar is enabled.

Small and large list boxes - Change the size of a control by specifying new values for the height and width.

Figure 16

## dSVG sample: ListView element

Default list view: (default attributes)

| | | |
|---|---|---|
| plu_1 | Apple | $1.27 |
| plu_2 | Banana | $0.59 |
| plu_3 | Orange | $1.99 |

Fixed-size list view: (rows=4)

| | | |
|---|---|---|
| plu_1 | Apple | $1.27 |
| plu_2 | Banana | $0.59 |
| plu_3 | Grapes | $2.19 |
| plu_4 | Kiwi | $0.89 |

Small list view: (h=25, w=100)

Large 4 column list view; (h=100, w=440)

| | | | |
|---|---|---|---|
| plu_1 | Apple | red | $1.27 |
| plu_2 | Banana | yellow | $0.59 |
| plu_3 | Grapes | purple | $2.19 |

***Note: The red outlines are not part of the controls. They are used to identify the dimensions for each control.**

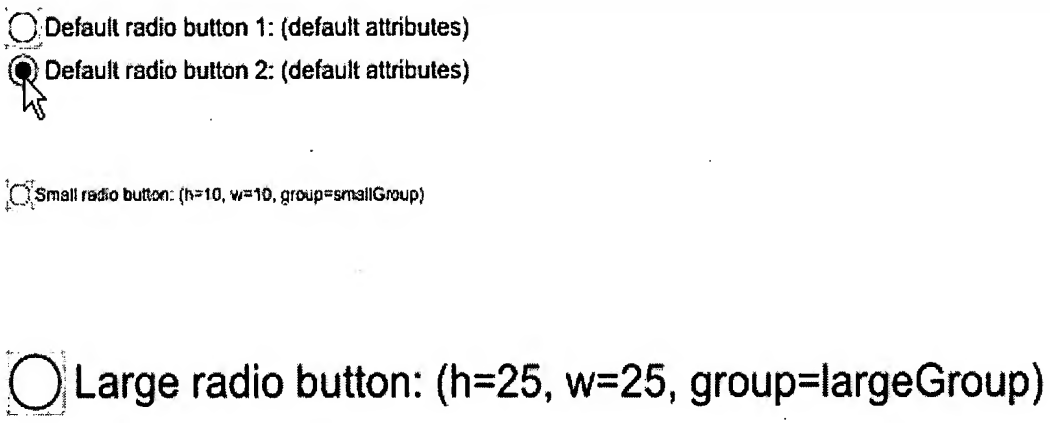
Default list view - This list view has 3 child item elements: Apple, Banana, and Orange.

Fixed-size list view - This list view displays a maximum of 4 items. If the number of items exceeds 4, a scrollbar is enabled.

Small and large list views - Change the size of the control by specifying new values for the height and width attributes.

Figure 17

## dSVG sample: SpinBox element




*Note: The red outlines are not part of the controls. They are used to identify the dimensions for each control.

---

Default, grouped radio buttons - Default radio buttons 1 and 2 belong to the same group.

Small and large radio buttons - Change the size of a control by specifying new values for the height and width attributes.

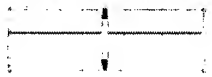
The default, small, and large radio buttons all belong to their own group. By default the group attribute is set to default.

Figure 18

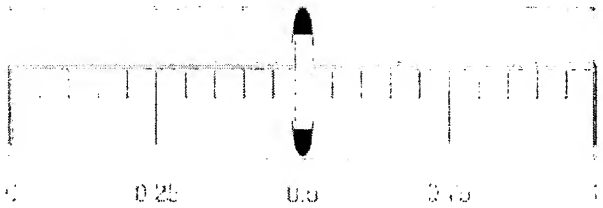
## dSVG sample:Slider element

Default slider: (default attributes)

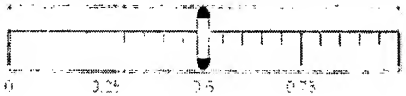

Small slider: (h=30, w=100)



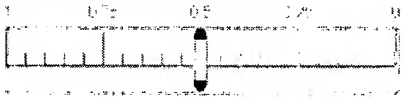
Large slider: (h=30, w=300)



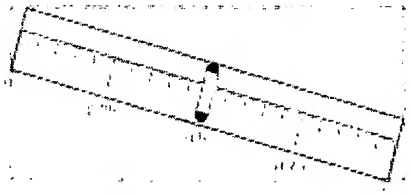
rotate=0, h=30, w=200



rotate=180, h=30, w=200



rotate=15, h=30, w=200



rotate=90

rotate=270

***Note: The red outlines are not part of the controls. They are used to identify the dimensions for each control.**

Default slider - Represents the default set of attributes.

Small and large sliders - Change the size of the control by specifying new values for the height and width attributes.

Rotated sliders - The rotate attribute is set the number of degrees specifed in the label.

Figure 19

## dSVG sample: SpinBox element

Default spin box: (default attributes)

1

Small spin box: (h=15, w=40, min= -10, max=0, value= -5, increment=1)

-5

Large spin box: (h=36, w=236, min= -100, max= 100, value=0, increment=10)

0

*Note: The red outlines are not part of the controls. They are used to identify the dimensions for each control.

---

Default spin box - This spin box has values of 1 to 10 in increments of 1. The initial value is 1.

Small and large spin boxes - Change the size of the control by specifying new values for the height and width attributes.

Figure 20

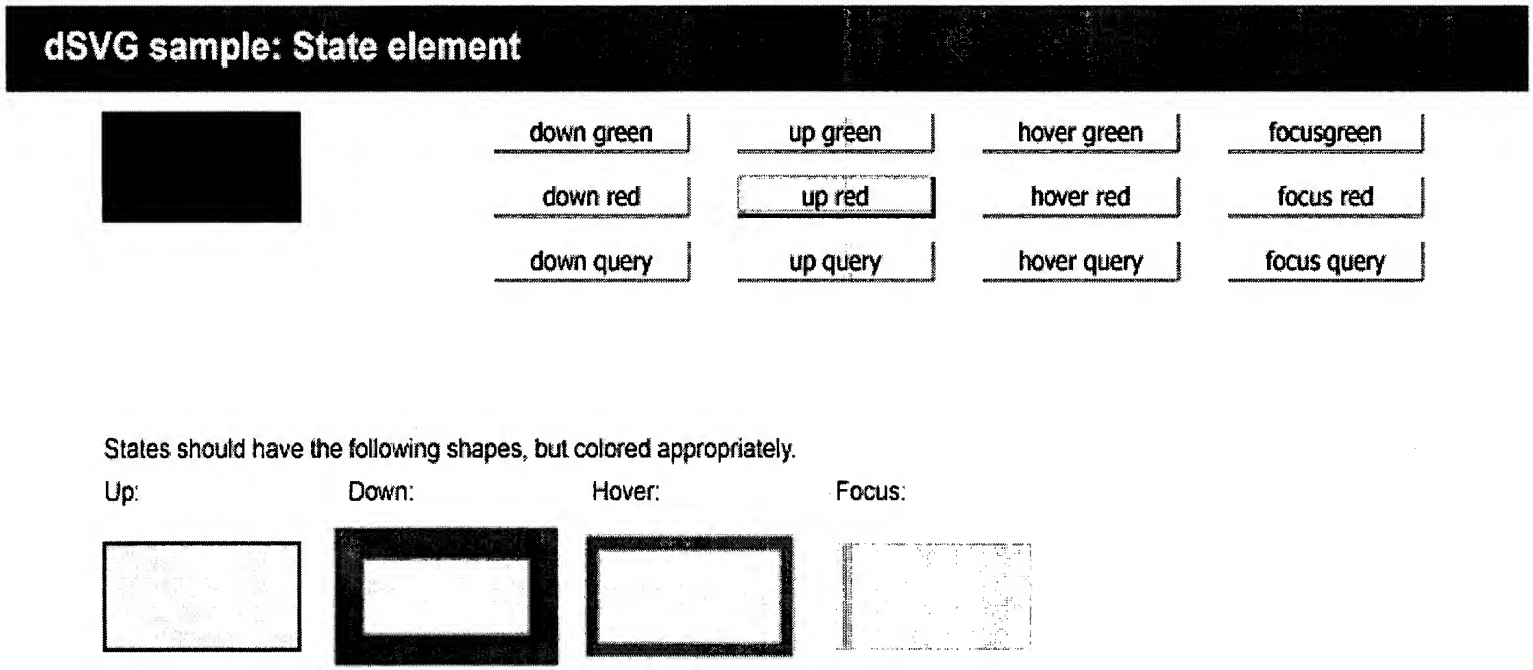
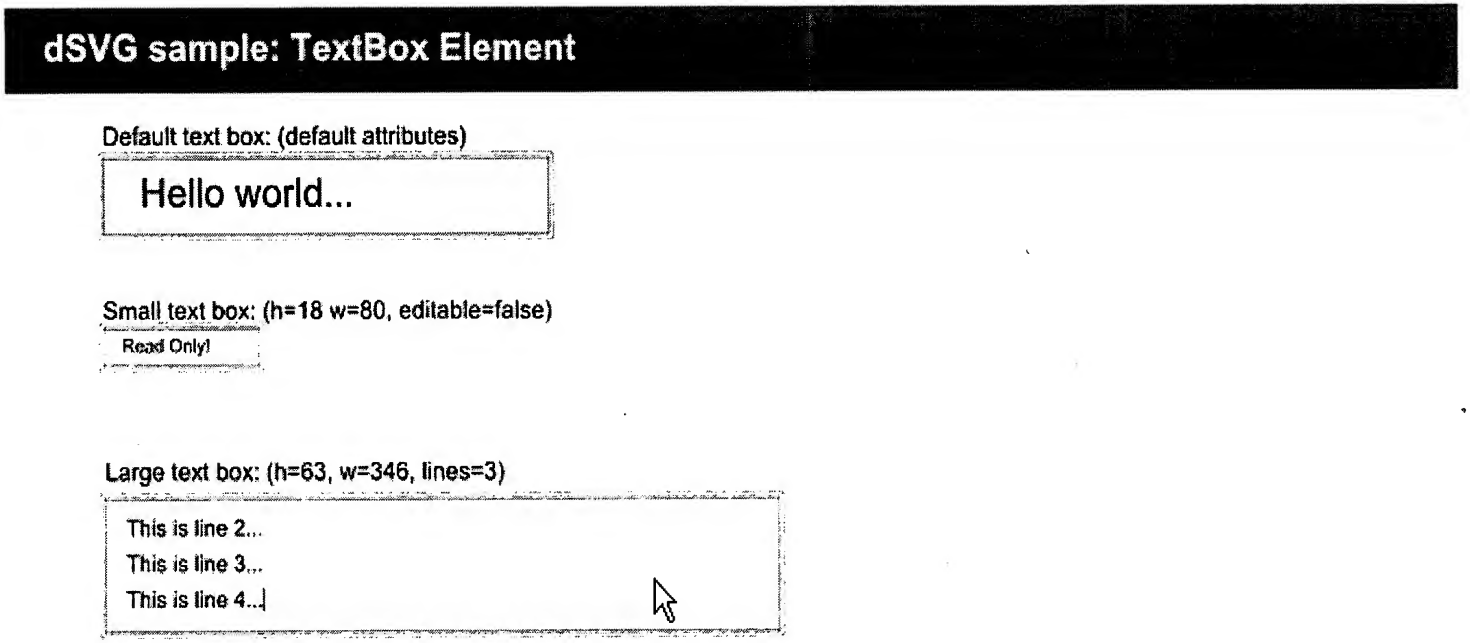

dSVG sample: State element
down green
up green
hover green
focusgreen
down red
up red
hover red
focus red
down query
up query
hover query
focus query
States should have the following shapes, but colored appropriately.
Up:
Down:
Hover:
Focus:
The state element defines an alternate skin to be available for the parent UI control.
This sample uses state to override the appearance of the custom button in the top left corner of the slide.
Pressing a query button returns the current state value to an alert.

Figure 21



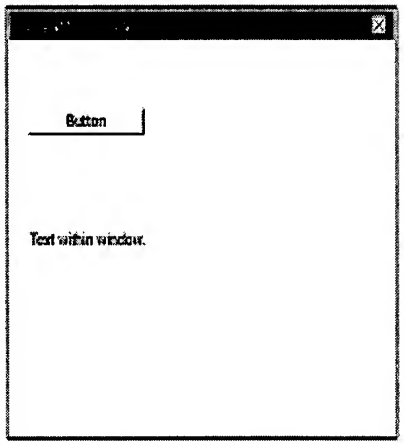

Default text box - This text box uses all of the default attribute settings.

Small text boxes - This text box has the editable attribute set to false which disables the entry of text into the text box.

Large text boxes - Change the size of a control by specifying new values for the height and width attributes.

Figure 22

## dSVG sample: Window element



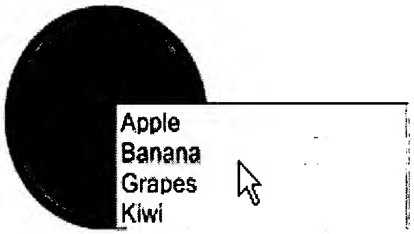

The window element defines a top-level container element that can be either modal or modeless, and can be moveable or not.

In this sample, the button element and the text element are added as children of the window.

Figure 23

## dSVG sample: ContextMenu element

Default Context Menu: (default attributes) - right-click within the circle and the Context Menu should appear.




---

Default context menu - Right-click within the circle to display the context menu.

The list consists of items which can be added directly as child elements of the context menu.

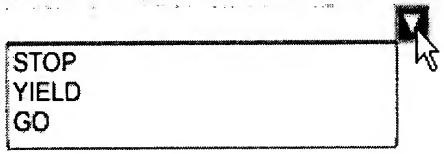
The context menu is associated with the circle by adding a dsvg:contextMenu attribute to the circle which references the context menu.

Figure 24

## dSVG sample: Share element

List box: (default attributes with the added attribute dsvg:share)


Combo box: (default attributes with the added attribute dsvg:share)



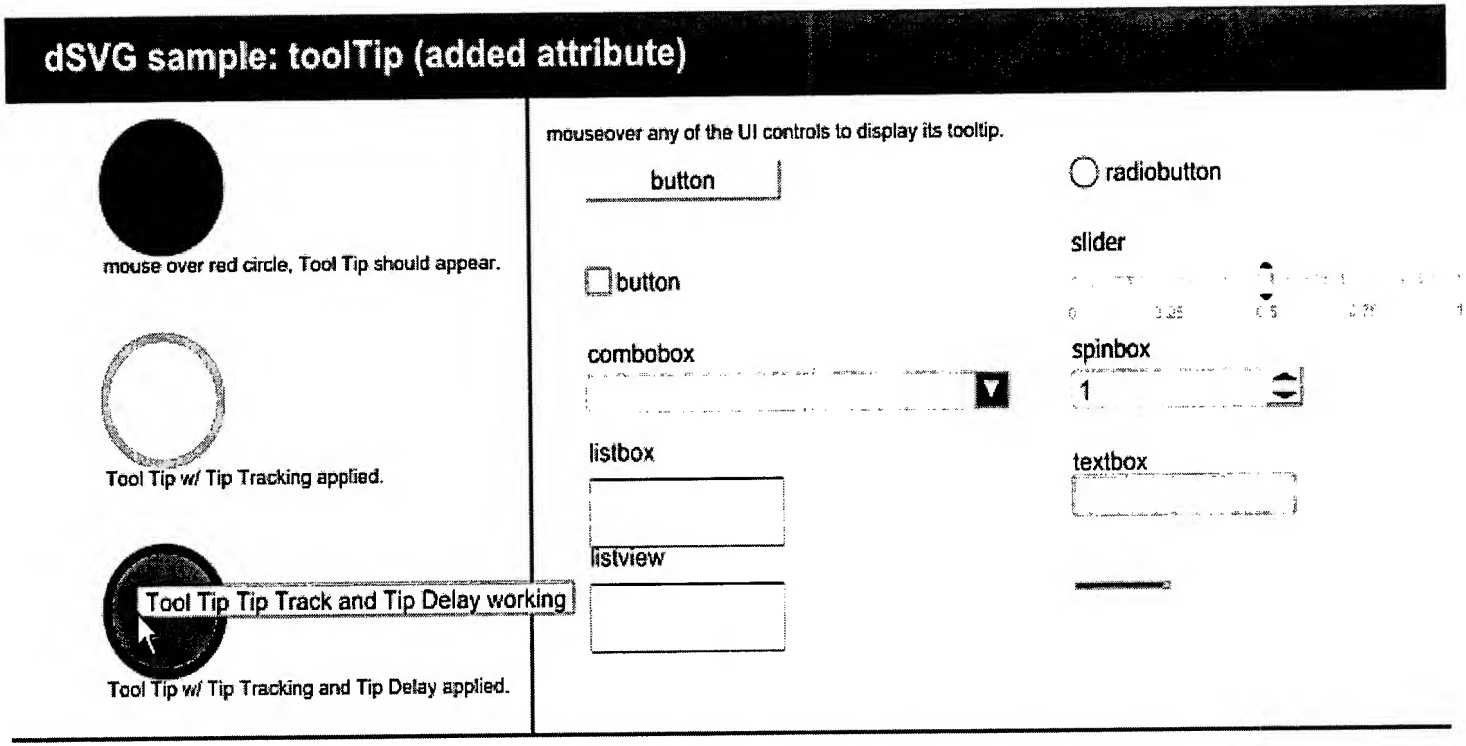

---

The share element is used to share a group of items with multiple elements.

This document shares the same set of items with the combo box and the list box.

Associate a share element with other elements by adding a dsvg:share attribute to the element that references the share element.

Figure 25




Content of file: dsvg:toolTip
The dsvg:toolTip attribute is applied to elements to enable the ability to display tooltips.
Tip tracking and Tip Delay are added as seperate attributes if desired.

Figure 26